Inside 64x0

Amlal El Mahrouss — NeKernel.org

Revision I

General information:

The NeKernel.org Open64x0 is a 128-bit RISC CPU, made with HPC computing in mind. It includes Out of order, Superscalar uArch called Harvard, you are reading it’s programming manual, It is assumed that you are familiar with the concept of registers, memory, instruction encoding and stack (FIFO, LIFO)

Register bank:

The NeKernel.org Open64x0 ISA consists of twenty general purpose registers and ten floating point registers.

Addressing modes:

The NeKernel.org Open64x0 has 3 instruction encoding: Register to Register, System call and Immediate.

They all start with the same format, opcode, funct3 and funct7, where these differ is after the (funct7), System call addressing takes its values from the stack, where as register to register addressing expects at least two registers within [0, 35], Immediate is quite different, for instance, use 'stw' to store a 64-bit word into a register. or 'lda' to load r0 from a register address. This helps to actually load useful stuff into registers.

The operating system is heavily encouraged to lookup for labels such as: (3:RuntimeSymbol:<Symbol>).

Figure 1: Register list of the X64000

Register name	Mnemonic
r0	Hardwired Zero Register
r1	Address Register 1
r2	Address Register 2
r3	Address Register 3
r4	Address Register 4
r5	Stack Register
r6	Data Register 1
r7	Data Register 2
r8	Data Register 3
r9	Data Register 4
r10	Temporary Register 1
r11	Temporary Register 2
r12	Temporary Register 3
r13	Temporary Register 4
r14	Temporary Register 5
r15	Condition Register 1
r16	Condition Register 2
r17	Program Counter Register
r18	Control Register
r19	Return Address register

Figure 2: Instruction encodings:

Figure 2.a: Register to Register encoding:

OPCODE	FUNCT3	FUNCT7	REG_LEFT	REG_RIGHT

Figure 2.b: Immediate encoding:

OPCODE	FUNCT3	FUNCT7	REG_LEFT	OFFSET	REG_RIGHT
OPCODE	FUNCT3	FUNCT7	REG	OFFSET	UNUSED
OPCODE	FUNCT3	FUNCT7	REG_LEFT	REG_RIGHT	OFFSET

Figure 2.c: Jump encoding:

OPCODE	FUNCT3	FUNCT7	OFFSET

What do they (funct3, funct7) even mean?

Opcode stands for operation code, it's the instruction you want to perform, whereas the funct3 tells it what it does, and funct7 what it is (that's btw how we tell what kind of encoding it uses).

For example take a load operation from 0x0000000 to r13, you need to make sure that you even want to load it. so opcode=0b0000011 and funct3=0b101.